

# MODUL

## ALUR KERJA SISTEM PERANGKAT LUNAK (DIAGRAM ALUR DATA)

### A. Tujuan Pembelajaran

1. Menjelaskan fungsi dan komponen DFD
2. Menjelaskan tingkatan level DFD
3. Menjelaskan spesifikasi proses DFD
4. Menjelaskan tahapan pembuatan DFD

### B. Materi Pembelajaran

#### Konsep Perancangan Terstruktur

Pendekatan perancangan terstruktur dimulai dari awal 1970. Pendekatan terstruktur dilengkapi dengan alat-alat (*tools*) dan teknik-teknik (*techniques*) yang dibutuhkan dalam pengembangan sistem, sehingga hasil akhir dari sistem yang dikembangkan akan diperoleh sistem yang strukturnya didefinisikan dengan baik dan jelas. Melalui pendekatan terstruktur, permasalahan yang kompleks di organisasi dapat dipecahkan dan hasil dari sistem akan mudah untuk dipelihara, fleksibel, lebih memuaskan pemakainya, mempunyai dokumentasi yang baik, tepat waktu, sesuai dengan anggaran biaya pengembangan, dapat meningkatkan produktivitas dan kualitasnya akan lebih baik (bebas kesalahan).

Data Flow Diagram (DFD) adalah alat pembuatan model yang memungkinkan profesional sistem untuk menggambarkan sistem sebagai suatu jaringan proses fungsional yang dihubungkan satu sama lain dengan alur data, baik secara manual maupun komputerisasi. DFD ini sering disebut juga dengan nama Bubble chart, Bubble diagram, model proses, diagram alur kerja, atau model fungsi. DFD ini adalah salah satu alat pembuatan model yang sering digunakan, khususnya bila fungsi-fungsi sistem merupakan bagian yang lebih penting dan kompleks dari pada data yang dimanipulasi oleh sistem. Dengan kata lain, DFD adalah alat pembuatan model yang memberikan penekanan hanya pada fungsi sistem. DFD ini merupakan alat perancangan sistem yang berorientasi pada alur data dengan konsep dekomposisi dapat digunakan untuk penggambaran analisa maupun rancangan sistem yang mudah dikomunikasikan oleh profesional sistem kepada pemakai maupun pembuat program. Tujuan DFD adalah :

1. Memberikan indikasi mengenai bagaimana data ditransformasi pada saat data bergerak melalui sistem.
2. Menggambarkan fungsi-fungsi(dan sub fungsi) yang mentransformasi aliran data.

#### Fungsi dan Komponen DFD

Fungsi dari data flow diagram adalah sebagai berikut :

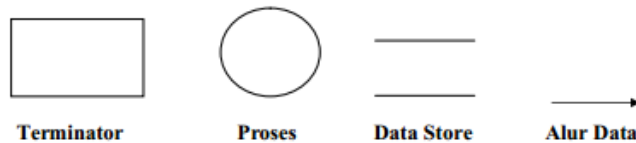
- Data Flow Diagram (DFD) adalah alat pembuatan model yang memungkinkan profesional sistem untuk menggambarkan sistem sebagai suatu jaringan proses

fungsi yang dihubungkan satu sama lain dengan alur data, baik secara manual maupun komputerisasi.

- DFD ini adalah salah satu alat pembuatan model yang sering digunakan, khususnya bila fungsi-fungsi sistem merupakan bagian yang lebih penting dan kompleks dari pada data yang dimanipulasi oleh sistem. Dengan kata lain, DFD adalah alat pembuatan model yang memberikan penekanan hanya pada fungsi sistem.
- DFD ini merupakan alat perancangan sistem yang berorientasi pada alur data dengan konsep dekomposisi dapat digunakan untuk penggambaran analisa maupun rancangan sistem yang mudah dikomunikasikan oleh profesional sistem kepada pemakai maupun pembuat program.

Sedangkan komponen dari data flow diagram adalah sebagai berikut :

**Menurut Yourdan dan DeMarco**



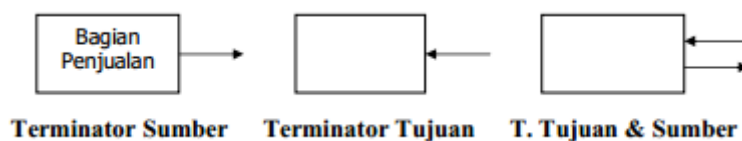
**Menurut Gene dan Serson**



1. Komponen Terminator / Entitas Luar

Terminator mewakili entitas eksternal yang berkomunikasi dengan sistem yang sedang dikembangkan. Biasanya terminator dikenal dengan nama entitas (*external entity*). Terdapat dua jenis terminator :

- Terminator Sumber (*source*) : merupakan terminator yang menjadi sumber.
- Terminator Tujuan (*sink*) : merupakan terminator yang menjadi tujuan data / informasi sistem.



Terminator dapat berupa orang, sekelompok orang, organisasi, departemen di dalam organisasi, atau perusahaan yang sama tetapi di luar kendali sistem yang sedang dibuat modelnya. Terminator dapat juga berupa departemen, divisi atau sistem di luar sistem yang berkomunikasi dengan sistem yang sedang dikembangkan. Komponen terminator ini perlu diberi nama sesuai dengan dunia luar yang berkomunikasi dengan sistem yang sedang dibuat modelnya, dan biasanya menggunakan kata benda, misalnya *Bagian Penjualan*, *Dosen*, *Mahasiswa*. Ada tiga hal penting yang harus diingat tentang terminator :

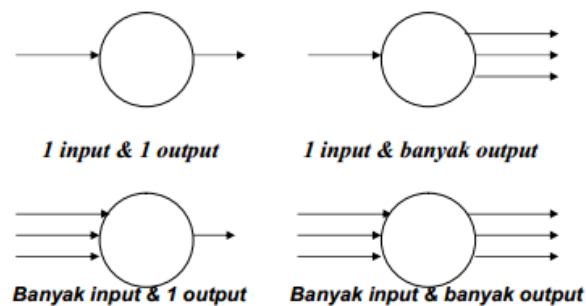
- Terminator merupakan bagian/lingkungan luar sistem. Alur data yang menghubungkan terminator dengan berbagai proses sistem, menunjukkan hubungan sistem dengan dunia luar.

- Profesional sistem tidak dapat mengubah isi atau cara kerja organisasi, atau prosedur yang berkaitan dengan terminator.
- Hubungan yang ada antar terminator yang satu dengan yang lain tidak digambarkan pada DFD.

## 2. Komponen Proses

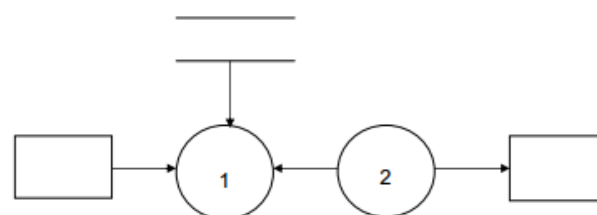
Komponen proses menggambarkan bagian dari sistem yang mentransformasikan input menjadi output. Proses diberi nama untuk menjelaskan proses/kegiatan apa yang

sedang/akan dilaksanakan. Pemberian nama proses dilakukan dengan menggunakan kata kerja transitif (kata kerja yang membutuhkan obyek), seperti *Menghitung Gaji*, *Mencetak KRS*, *Menghitung Jumlah SKS*. Ada empat kemungkinan yang dapat terjadi dalam proses sehubungan dengan input dan output :



Ada beberapa hal yang perlu diperhatikan tentang proses :

- Proses harus memiliki input dan output.
- Proses dapat dihubungkan dengan komponen terminator, data store atau proses melalui alur data.
- Sistem/bagian/divisi/departemen yang sedang dianalisis oleh profesional sistem digambarkan dengan komponen proses. Berikut ini merupakan suatu contoh proses yang salah :



Gambar 1. Contoh proses

Umumnya kesalahan proses di DFD adalah :

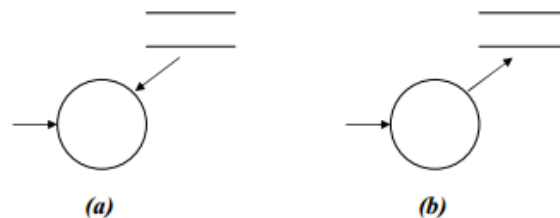
- Proses mempunyai input tetapi tidak menghasilkan output. Kesalahan ini disebut dengan **black hole** (lubang hitam), karena data masuk ke dalam proses dan lenyap tidak berbekas seperti dimasukkan ke dalam lubang hitam (*lihat proses 1*).
- Proses menghasilkan output tetapi tidak pernah menerima input. Kesalahan ini disebut dengan **miracle** (ajaib), karena ajaib dihasilkan output tanpa pernah menerima input (*lihat proses 2*).

## 3. Komponen Data Store

Komponen ini digunakan untuk membuat model sekumpulan paket data dan diberi nama dengan kata benda jamak, misalnya *Mahasiswa*. Data store ini biasanya berkaitan dengan penyimpanan, seperti file atau database yang berkaitan dengan penyimpanan secara komputerisasi, misalnya file disket, file harddisk, file pita magnetik. Data store juga berkaitan dengan penyimpanan secara manual seperti buku alamat, file folder, dan agenda. Suatu data store dihubungkan dengan alur data **hanya pada komponen proses**, tidak dengan komponen DFD lainnya. Alur data yang menghubungkan data store dengan suatu proses mempunyai pengertian sebagai berikut :

- a. **Alur data dari data store** yang berarti sebagai pembacaan atau pengaksesan satu paket tunggal data, lebih dari satu paket data, sebagian dari satu paket tunggal data, atau sebagian dari lebih dari satu paket data untuk suatu proses (*lihat gambar 2 (a)*).
- b. **Alur data ke data store** yang berarti sebagai pengupdatean data, seperti menambah satu paket data baru atau lebih, menghapus satu paket atau lebih, atau mengubah/memodifikasi satu paket data atau lebih (*lihat gambar 2 (b)*).

Pada pengertian pertama jelaslah bahwa data store tidak berubah, jika suatu paket data/informasi berpindah dari data store ke suatu proses. Sebaliknya pada pengertian kedua data store berubah sebagai hasil alur yang memasuki data store. Dengan kata lain, proses alur data bertanggung jawab terhadap perubahan yang terjadi pada data store.



Gambar 2. Implementasi data store

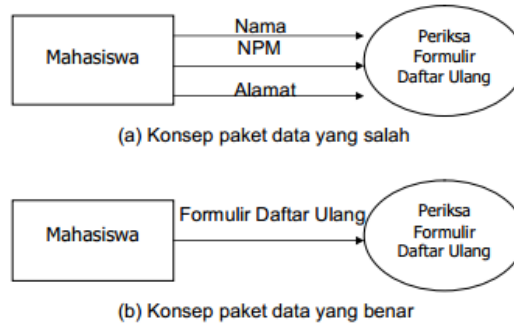
#### 4. Komponen Data Flow / Alur Data

Suatu data flow / alur data digambarkan dengan anak panah, yang menunjukkan arah menuju ke dan keluar dari suatu proses. Alur data ini digunakan untuk menerangkan perpindahan data atau paket data/informasi dari satu bagian sistem ke bagian lainnya. Selain menunjukkan arah, alur data pada model yang dibuat oleh profesional sistem dapat merepresentasikan bit, karakter, pesan, formulir, bilangan real, dan macam-macam informasi yang berkaitan dengan komputer. Alur data juga dapat merepresentasikan data/informasi yang tidak berkaitan dengan komputer.

- a. **Alur data** perlu **diberi nama** sesuai dengan data/informasi yang dimaksud, biasanya pemberian nama pada alur data dilakukan dengan menggunakan **kata benda**, contohnya **Laporan Penjualan**. Ada empat konsep yang perlu diperhatikan dalam penggambaran alur data, yaitu :

- 1) **Konsep Paket Data (Packets of Data)**

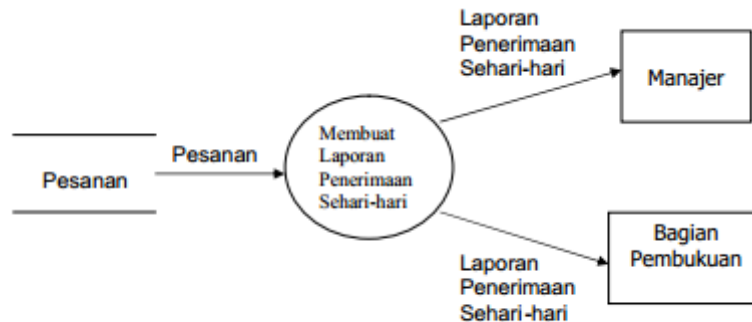
Apabila dua data atau lebih mengalir dari *suatu sumber yang sama* menuju ke *tujuan yang sama* dan mempunyai hubungan, dan harus dianggap sebagai satu alur data tunggal, karena data itu mengalir bersama-sama sebagai satu paket.



Gambar 3. Konsep paket data

## 2) Konsep Alur Data Menyebar (*Diverging Data Flow*)

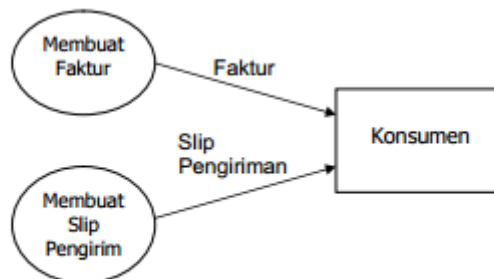
Alur data menyebar menunjukkan sejumlah tembusan paket data yang berasal dari *sumber yang sama* menuju ke *tujuan yang berbeda*, atau paket data yang kompleks dibagi menjadi beberapa elemen data yang dikirim ke tujuan yang berbeda, atau alur data ini membawa paket data yang memiliki nilai yang berbeda yang akan dikirim ke tujuan yang berbeda.



Gambar 4. Konsep alur data menyebar

## 3) Konsep Alur Data Mengumpul (*Converging Data Flow*)

Beberapa alur data yang *berbeda sumber* bergabung bersamasama menuju ke *tujuan yang sama*.

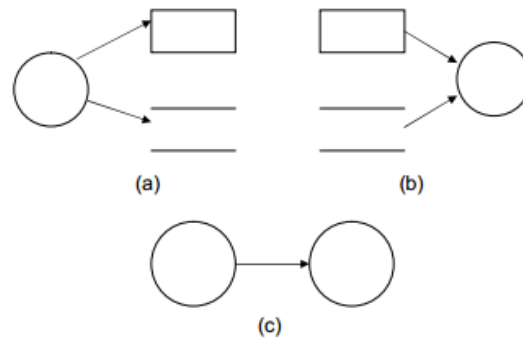


Gambar 5. Konsep alur data mengumpul

## 4) Konsep Sumber atau Tujuan Alur Data

Semua alur data harus *minimal mengandung satu proses*. Maksud kalimat ini adalah :

- Suatu alur data dihasilkan dari suatu *proses* dan menuju ke *suatu data store* dan/atau *terminator* (*lihat gambar 6 (a)*).
- Suatu alur data dihasilkan dari suatu *data store* dan/atau *terminator* dan menuju ke suatu *proses* (*lihat gambar 6 (b)*)
- Suatu alur data dihasilkan dari suatu *proses* dan menuju ke suatu *proses* (*lihat gambar 6 (c)*).



Gambar 6. Konsep sumber atau tujuan alur data

### **Bentuk Data Flow Diagram**

Terdapat dua bentuk DFD, yaitu **Diagram Alur Data Fisik**, dan **Diagram Alur data Logika**. Diagram alur data fisik lebih menekankan pada bagaimana proses dari sistem diterapkan, sedangkan diagram alur data logika lebih menekankan proses-proses apa yang terdapat di sistem.

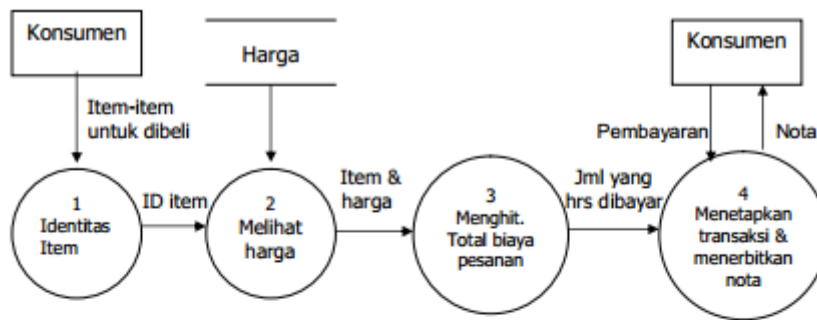
#### 1. Diagram Alur Data Fisik (DADF)

DADF lebih tepat digunakan untuk menggambarkan sistem yang ada (sistem yang lama). Penekanan dari DADF adalah bagaimana proses-proses dari sistem diterapkan (dengan cara apa, oleh siapa dan dimana), termasuk proses-proses manual. Untuk memperoleh gambaran bagaimana sistem yang ada diterapkan, DADF harus memuat :

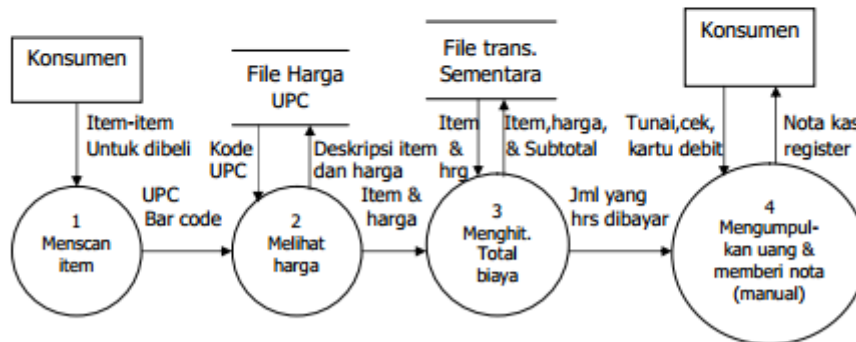
- Proses-proses manual juga digambarkan.
- Nama dari alur data harus memuat keterangan yang cukup terinci untuk menunjukkan bagaimana pemakai sistem memahami kerja system.
- Simpanan data dapat menunjukkan simpanan non komputer.
- Nama dari simpanan data harus menunjukkan tipe penerapannya apakah secara manual atau komputerisasi. Secara manual misalnya dapat menunjukkan buku catatan, meja pekerja. Sedang cara komputerisasi misalnya menunjukkan file urut, file database.
- Proses harus menunjukkan nama dari pemroses, yaitu orang, departemen, sistem komputer, atau nama program komputer yang mengakses proses tersebut.

#### 2. Diagram Alur Data Logika (DADL)

DADL lebih tepat digunakan untuk menggambarkan sistem yang akan diusulkan (sistem yang baru). Untuk sistem komputerisasi, penggambaran DADL hanya menunjukkan kebutuhan proses dari sistem yang diusulkan secara logika, biasanya proses-proses yang digambarkan hanya merupakan proses-proses secara komputer saja.



(a) Diagram Alur Data Logika



(b) Diagram Alur Data Fisik

Gambar 7. DADF dan DADL

### Tingkatan Level DFD

#### 1. Diagram Konteks

Menggambarkan satu lingkaran besar yang dapat mewakili seluruh proses yang terdapat di dalam suatu sistem. Merupakan tingkatan tertinggi dalam DFD dan biasanya diberi nomor 0 (nol). Semua entitas eksternal yang ditunjukkan pada diagram konteks berikut aliran-aliran data utama menuju dan dari sistem. Diagram ini sama sekali tidak memuat penyimpanan data dan tampak sederhana untuk diciptakan.

#### 2. Diagram Nol (diagram level-1)

Merupakan satu lingkaran besar yang mewakili lingkaran-lingkaran kecil yang ada di dalamnya. Merupakan pemecahan dari diagram Konteks ke diagram Nol. di dalam diagram ini memuat penyimpanan data.

#### 3. Diagram Rinci

Merupakan diagram yang menguraikan proses apa yang ada dalam diagram Nol.

### Spesifikasi Proses

Syarat pembuatan DFD ini akan menolong profesional sistem untuk menghindari pembentukan DFD yang salah atau DFD yang tidak lengkap atau tidak konsisten secara logika. Beberapa syarat pembuatan DFD dapat menolong profesional sistem untuk membentuk DFD yang benar, menyenangkan untuk dilihat dan mudah dibaca oleh pemakai. Syarat-syarat pembuatan DFD ini adalah :

#### 1. Pemberian nama untuk tiap komponen DFD

2. Pemberian nomor pada komponen proses
3. Penggambaran DFD sesering mungkin agar enak dilihat
4. Penghindaran penggambaran DFD yang rumit
5. Pemastian DFD yang dibentuk itu konsisten secara logika

### **Tahapan Pembuatan DFD**

Tips dalam membuat DFD:

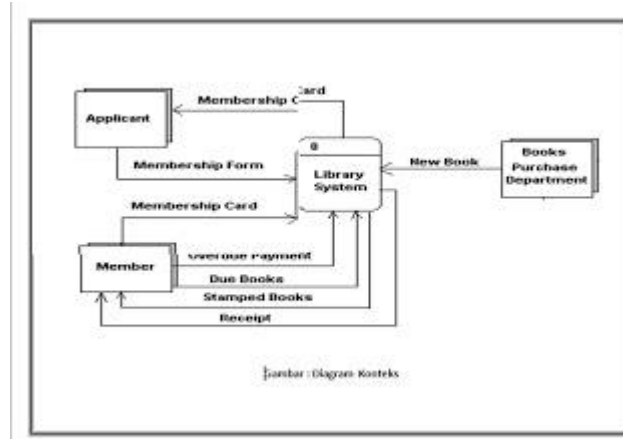
1. Pilih notasi sehingga proses yang didekomposisi atau tidak didekomposisi dapat dibaca dengan mudah.
2. Nama proses harus terdiri dari kata kerja dan kata benda.
3. Nama yang dipakai untuk proses, data store, dataflow harus konsisten (identitas perlu).
4. Setiap level harus konsisten aliran datanya dengan level sebelumnya.
5. Usahakan agar external entity pada setiap level konsisten peletakkannya.
6. Banyaknya proses yang disarankan pada setiap level tidak melebihi 7 proses.
7. Dekomposisi berdasarkan kelompok data lebih disarankan (memudahkan aliran data ke storage yang sama).
8. Nama Proses yang umum hanya untuk proses yang masih akan didekomposisi.
9. Pada Proses yang sudah tidak didekomposisi, nama Proses dan nama Data harus sudah spesifik.
10. Aliran ke storage harus melalui proses, tidak boleh langsung dari external entity.
11. Aliran data untuk Proses Report .. : harus ada aliran keluar. Akan ada aliran masuk jika perlu parameter untuk mengaktifkan report. Aliran data yang tidak ada data storenya harus diteliti, apakah memang tidak mencerminkan persisten entity (perlu disimpan dalam file/tabel), yaitu kelak hanya akan menjadi variabel dalam program.

Langkah membuat/menggambar DFD

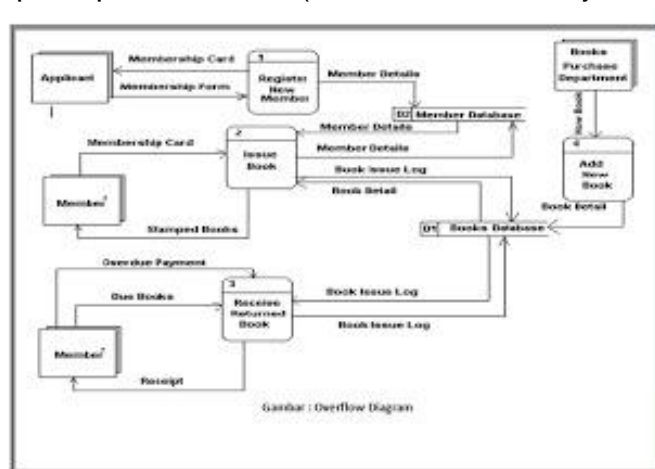
Tidak ada aturan baku untuk menggambarkan DFD. Tapi dari berbagai referensi yang ada, secara garis besar langkah untuk membuat DFD adalah :

1. Identifikasi Entitas Luar, Input dan Output
2. Identifikasi terlebih dahulu semua entitas luar, input dan output yang terlibat di sistem.
3. Buat Diagram Konteks (diagram context). Diagram ini adalah diagram level tertinggi dari DFD yang menggambarkan hubungan sistem dengan lingkungan luarnya. Caranya :
  - ✓ Tentukan nama sistemnya.
  - ✓ Tentukan batasan sistemnya.
  - ✓ Tentukan terminator apa saja yang ada dalam sistem.
  - ✓ Tentukan apa yang diterima/diberikan external entity dari/ke sistem.
  - ✓ Gambarkan diagram konteks.





4. Buat Diagram Level Zero (Overview Diagram). Diagram ini adalah dekomposisi dari diagram konteks. Caranya :
- Tentukan proses utama yang ada pada sistem.
  - Tentukan apa yang diberikan/diterima masing-masing proses ke/dari sistem sambil memperhatikan konsep keseimbangan (alur data yang keluar/masuk dari suatu level harus sama dengan alur data yang masuk/keluar pada level berikutnya).
  - Apabila diperlukan, munculkan data store (master) sebagai sumber maupun tujuan alur data.
  - Hindari perpotongan arus data.
  - Beri nomor pada proses utama (nomor tidak menunjukkan urutan proses).

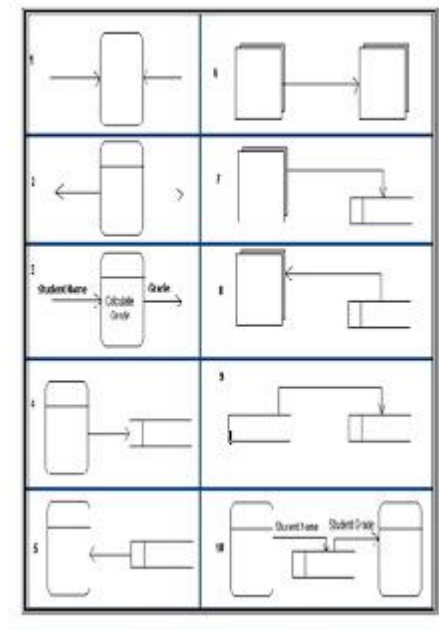


5. Buat Diagram Level Satu. Diagram ini merupakan dekomposisi dari diagram level zero. Caranya :
- Tentukan proses yang lebih kecil (sub-proses) dari proses utama yang ada di level zero.
  - Tentukan apa yang diberikan/diterima masing-masing sub-proses ke/dari sistem dan perhatikan konsep keseimbangan.
  - Apabila diperlukan, munculkan data store (transaksi) sebagai sumber maupun tujuan alur data.
  - Hindari perpotongan arus data.

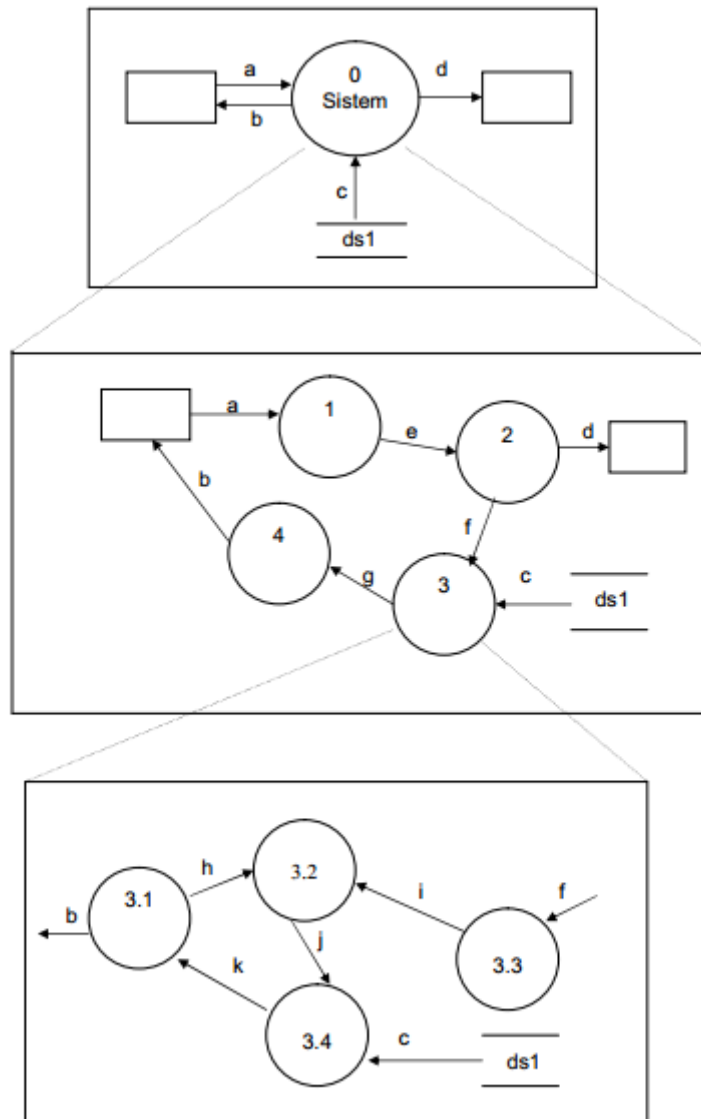
- Beri nomor pada masing-masing sub-proses yang menunjukkan dekomposisi dari proses sebelumnya. Contoh : 1.1, 1.2, 2

### **Kesalahan dalam Pembuatan DFD**

Umumnya kesalahan dalam pembuatan DFD adalah :



1. Proses mempunyai input tetapi tidak menghasilkan output. Kesalahan ini disebut dengan black hole (lubang hitam), karena data masuk ke dalam proses dan lenyap tidak berbekas seperti dimasukkan ke dalam lubang hitam.
2. Proses menghasilkan output tetapi tidak pernah menerima input. Kesalahan ini disebut dengan miracle (ajaib), karena ajaib dihasilkan output tanpa pernah menerima input.
3. Input yang masuk tidak sesuai dengan kebutuhan proses.
4. Data Store tidak memiliki keluaran.
5. Data Store tidak memiliki masukan.
6. Hubungan langsung antar entitas luar.
7. Masukan langsung entitas data store.
8. Keluaran langsung dari data store ke Entitas luar.
9. Hubungan langsung antar data store.
10. Data masukan dan keluaran yang tidak bersesuaian dalam data store.



Gambar 11. Levelisasi DFD

**C. Tugas Pembelajaran**

1. Sebutkan dan jelaskan dua bentuk Data Flow Diagram (DFD) !
2. Apa saja yang termasuk dalam tingkatan level DFD ? Jelaskan !
3. Jelaskan dari syarat-syarat pembuatan DFD !
4. Sebutkan tiga hal penting tentang terminator !

**D. Referensi**

Pemodelan Perangkat Lunak, Diagram Alur Data. Rekayasa Perangkat Lunak, SMK Negeri 8 Malang